Woods Hole Oceanographic Institution

1930

Dominant Run-Length Method for Image Classification

by

Xiaoou Tang

June 1997

Technical Report

19970924 013

DTIC QUALITY INSPECTED 4

WHOI-97-07

Dominant Run-Length Method for Image Classification

by

Xiaoou Tang

Woods Hole Oceanographic Institution
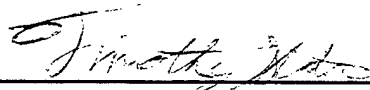Woods Hole, Massachusetts 02543

June 1997

Technical Report

Approved for public release; distribution unlimited.

Approved for Distribution:

Timothy Stanton, Chair

Department of Applied Ocean Physics and Engineering

Dominant Run-Length Method For Image Classification

Xiaoou Tang

Department of Applied Ocean Physics and Engineering
Woods Hole Oceanographic Institution
Woods Hole, MA 02543

Abstract -- In this paper, we develop a new run-length texture feature extraction algorithm that significantly improves image classification accuracy over traditional techniques. By directly using part or all of the run-length matrix as a feature vector, much of the texture information is preserved. This approach is made possible by the introduction of a new multi-level dominant eigenvector estimation algorithm. It reduces the computational complexity of the Karhunen-Loeve Transform by several orders of magnitude. Combined with the Bhattacharyya distance measure, they form an efficient feature selection algorithm. The advantage of this approach is demonstrated experimentally by the classification of two independent texture data sets. Perfect classification is achieved on the first data set of eight Brodatz textures. The 97% classification accuracy on the second data set of sixteen Vistex images further confirms the effectiveness of the algorithm. Based on the observation that most texture information is contained in the first few columns of the run-length matrix, especially in the first column, we develop a new fast, parallel run-length matrix computation scheme. Comparisons with the co-occurrence and wavelet methods demonstrate that the run-length matrices contain great discriminatory information and that a method of extracting such information is of paramount importance to successful classification.

1

I. INTRODUCTION

Texture is the term used to characterize the surface of a given object or region and it is undoubtedly one of the main features utilized in image processing and pattern recognition. Texture analysis plays a fundamental role in classifying objects and segmenting the significant regions of a given image. A solution to the texture analysis problem will greatly advance the image processing and pattern recognition fields and bring great benefit to many possible industrial applications [29]. However, the diversity of natural and artificial textures makes it difficult to give a universal definition of texture, resulting in a large number of texture analysis techniques.

A good survey of traditional statistical texture analysis methods was given in [29]. They include the spatial gray level dependence method (SGLDM) [13] [14], the gray level run length method (GLRLM) [4] [7] [12], the gray level difference method (GLDM) [30], and the power spectrum method (PSM) [17] [21]. SGLDM is based on the estimation of the second-order joint probability density functions (PDF) of the gray levels of two pixels separated by a distance d in a direction α. It is the most widely used texture analysis method due to its consistently superior performance over the other three methods. The GLRLM estimates the PDF of the gray level run lengths of texture. The GLDM method uses functions of the first order PDF of the gray level difference of two nearby pixels to compute texture features, while the PSM method studies the power spectrum statistics in the frequency domain. The performance ranking of the four methods from good to poor are generally, SGLDM, GLDM, PSM and GLRLM.

A new multi-channel approach called "texture energy analysis" was first introduced by Laws [20]. Laws used a set of small empirical filter masks to filter the texture image, then computed the variances in each channel output as the texture features. The shapes of the filter masks are similar to directional edge detectors. Later Ade [1] and Unser [27] [28] developed the eigenfilter approach, with Laws' empirical filter banks replaced by the eigenvectors of the covariance matrix of the local texture neighborhood vectors. Both the Laws filter and the eigenfilter approaches are shown to have texture classification capability comparable to that of the co-occurrence method.

Almost parallel to the development of the eigenfilter theory, the Gabor filter became increasingly used in designing texture analysis algorithms [9] [10] [16] [22]. Jain et al. [16] and Dunn et al. [9] developed several filter design procedures using Gabor functions. Malik and Perona [22] derived a filter bank combination structure mimicking the human early vision system, which perhaps has provided the most detailed justification for a particular filter-bank structure [9].

However, the filter outputs of the above multichannel approaches are not orthogonal, thus leading to a large overcomplete representation of the original image. Recent advances in wavelet [8] [23] [24] [25] and wavelet packet theory [5] [25] provide a promising solution for this problem. The texture research community is currently devoting considerable effort to wavelet applications in texture analysis [3] [15] [18]. Henke-Reed and Cheng [15] performed a wavelet transform to texture images, using the energy ratios between frequency channels as the features. Chang and Kuo [3] developed a tree structured wavelet transform algorithm for texture classification and segmentation, which is similar to the wavelet packet best basis selection algorithm of Coifman and Wickerhauser [5]. Both the standard wavelet features and

3

the wavelet packet energy features were used directly as texture features by Laine and Fan [18] in their texture classification work.

Many comparison studies have been conducted for various statistical texture analysis methods. Weszka *et al.* [30] experimentally compared features on terrain images and found that the co-occurrence features were best among those studied, ranking ahead of the GLDM, the PSM and the run-length method. Conners and Harlow [6] compared features on generated textures and drew similar conclusions. Unser [27] showed that the eigenfilter features gave texture classification performance comparable to that of the co-occurrence features. Wavelet features have been demonstrated by Chang and Kuo [3] to give performance similar to that obtained by the eigenfilter features. One definite conclusion that can be drawn from these and many other texture studies is that the run-length features are the least efficient texture features. The applications of the run-length method have been very limited compared to other approaches, since introduced by Galloway [12].

In this paper we investigate this least used method from a new approach. By using a new multi-level dominant eigenvector estimation algorithm and the Bhattacharyya distance measure for texture feature selection, we demonstrate that texture features extracted from the run-length matrix can give great classification results. We then experimentally compare the new run-length method with the widely used co-occurrence method and the recently proposed wavelet method.

This paper is organized into four sections. Section II introduces the original definition of the run-length matrix and several of its variations, then reviews the traditional run-length features and describes the new run-length feature extraction algorithm. Section III presents the texture classification experimental results. The conclusions are summarized in Section IV.

II. METHODOLOGY

A. Definition of the run-length matrices.

With the observation that, in a coarse texture, relatively long gray-level runs would occur more often and that a fine texture should contain primarily short runs, Galloway proposed the use of a run-length matrix for texture feature extraction [12]. For a given image, a run-length matrix $p(i, j)$ is defined as the number of runs with pixels of gray level i and run length j. An example of the run-length matrices is shown in Fig.1, where four directional run-length matrices are computed from the original image. Various texture features can then be derived from these matrices.

In this paper, we design several new run-length matrices, which are slight but unique variations of the traditional run-length matrix. For a run-length matrix $p(i, j)$, let M be the number of gray levels and N be the maximum run length. The four new matrices are defined as follows.

Gray Level Run Length Pixel Number Matrix - GLRLPNM:

$$p_p(i, j) = p(i, j) \cdot j. \tag{1}$$

Each element of the matrix represents the number of pixels of run-length j and gray-level i. Compared to the original matrix, the new matrix gives equal emphasis to all length of runs in an image.

Gray Level Run Number Vector - GLRNV:

$$p_g(i) = \sum_{j=1}^{N} p(i, j).$$ (2)

This vector represents the sum distribution of the number of runs with gray level i.

Run Length Run Number Vector - RLRNV:

$$p_r(j) = \sum_{i=1}^{M} p(i, j).$$ (3)

This vector represents the sum distribution of the number of runs with run length j.

Gray Level Run-Length-One Vector - GLRLOV:

$$p_o(i) = p(i, 1).$$ (4)

Figure 2 shows the four directional run-length matrices of several natural texture samples. Notice that the first column of each of the four directional run-length matrices is overwhelmingly larger than the other columns. This may mean that most texture information is contained in the run-length-one vector. The advantages of using this vector are that it offers significant feature length reduction and that a fast parallel run-length matrix computation can replace the conventional serial searching algorithm. For example, the positions of pixels with run length one in the horizontal direction can be found by a logical "and" operation on the outputs of the forward and backward derivative of the original image:

$$f(i, j) = x(i, j) - x(i, j-1),$$ (5)

$$b(i, j) = x(i, j-1) - x(i, j),$$ (6)

$$o(i, j) = f(i, j) \cap b(i, j),$$ (7)

where $x(i, j)$ is the texture image whose pixels outside the image boundary are set to zero, and $\cap$ represents the logical "and" operation. Then $p_o(i)$ can be obtained by computing the histogram of $x(i, j)_{o(i, j) = 1}$. To find the starting pixel position for runs with length two, a similar scheme can be employed,

$$f_2(i, j) = (f(i, j) \neq 0) - o(i, j),$$ (8)

$$b_2(i, j) = (b(i, j) \neq 0) - o(i, j),$$ (9)

$$o_2(i, j) = f_2(i, j) \cap b_2(i, j+1).$$ (10)

In fact, the gray level run number vector $p_g(i)$ can also be obtained with the above approach by

computing the histogram of $x(i, j)_{f(i, j) \neq 0}$.

The matrix and vectors defined above are not designed for the extraction of traditional features. Along with the original run-length matrix, they are used in the new feature extraction approach in section II-C. The next section gives a review of the traditional feature extraction.

B. Traditional run-length features

From the original run-length matrix $p(i, j)$, many numerical texture measures can be computed. The five original features of run-length statistics derived by Galloway [12] are:

Short Run Emphasis (SRE)

$$SRE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} p(i, j)/j^2 = \frac{1}{n_r} \sum_{j=1}^{N} p_r(j)/j^2, \tag{11}$$

Long Run Emphasis (LRE)

$$LRE = \frac{1}{n_r} \sum_{i=1}^{M} \sum_{j=1}^{N} p(i, j) \cdot j^2 = \frac{1}{n_r} \sum_{j=1}^{N} p_r(j) \cdot j^2, \tag{12}$$

Gray Level Nonuniformity (GLN)

$$GLN = \frac{1}{n_r} \sum_{i=1}^{M} \left(\sum_{j=1}^{N} p(i, j) \right)^2 = \frac{1}{n_r} \sum_{i=1}^{M} p_g(i)^2, \tag{13}$$

Run Length Nonuniformity (RLN)

$$GLN = \frac{1}{n_r} \sum_{j=1}^{N} \left(\sum_{i=1}^{M} p(i, j) \right)^2 = \frac{1}{n_r} \sum_{j=1}^{N} p_r(j)^2, \tag{14}$$

Run Percentage

$$RP = \frac{n_r}{n_p}, \tag{15}$$

where n_r is the total number of runs and n_p is the number of pixels in the image. Based on the observation that most features are only functions of $p_r(j)$, without considering the gray level information contained in $p_g(i)$, Chu et al. [4] proposed two new features,

Low Gray-level Run Emphasis (LGRE)

$$LGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j)/i^2 = \frac{1}{n_r}\sum_{i=1}^{M} p_g(i)/i^2, \tag{16}$$

High Gray-level Run Emphasis (HGRE)

$$HGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j)\cdot i^2 = \frac{1}{n_r}\sum_{i=1}^{M} p_g(i)\cdot i^2, \tag{17}$$

to extract gray level information in the matrix. In a more recent study, Dasarathy and Holder [7] described another four feature extraction functions following the idea of joint statistical measure of gray level and run length,

Short Run Low Gray-level Emphasis (SRLGE)

$$SRLGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j)/(i^2\cdot j^2), \tag{18}$$

Short Run High Gray-level Emphasis (SRHGE)

$$SRHGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j)\cdot i^2/j^2, \tag{19}$$

Long Run Low Gray-level Emphasis (LRLGE)

$$LRLGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j)\cdot j^2/i^2, \tag{20}$$

Long Run High Gray-level Emphasis (LRHGE)

$$LRHGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j)\cdot i^2\cdot j^2. \tag{21}$$

Dasarathy and Holder [7] tested all eleven features on the classification of a set of cell images and showed that the last four features gave much better performance. However, the data set they used was small, with only 20 samples in each of the four image classes. In Section III, we test these features on a much larger data set with 225 samples in each of eight image classes.

These features are all based on intuitive reasoning, in an attempt to capture some apparent properties of run-length distribution. For example, the eight features illustrated in Fig. 3 are weighted-sum measures of

7

the run-length concentration in the eight directions, i.e., the positive and negative 0-, 90-, 45-, and 135-degree directions. Similar to the way in which these features are derived, we could define more *ad hoc* features. Two drawbacks of this approach kept us from doing so: there is no theoretical proof that, given a certain number of features, maximum texture information can be extracted from the run-length matrix, and many of these features are highly correlated with each other. For example, for an image with high long-run emphasis, the short-run emphasis must be relatively small, so the long-run-emphasis features and the short-run-emphasis features essentially measure the same texture property.

C. Dominant run-length method (DRM)

Instead of developing new functions to extract texture information, we use the run-length matrix as the texture feature vector directly to preserve all information in the matrix. However, this again introduces two problems: the large dimensionality of the feature vector and the high-degree correlation of the neighborhood features.

To alleviate the first problem, observe the run-length matrix in Fig. 2 more closely. We see that most non-zero values concentrate in the first few columns of the matrix. Moreover the information in these first few columns, i. e., the short-run section, is correlated with that of the rest of the matrix, i. e., the long-run section, because for each row of the run-length matrix an image with a high long-run value will have a smaller short-run value. By using only the first few columns as the feature vector, the information in the long run section is not simply discarded but is mostly preserved in the feature vector. Another advantage of using only the first few columns is that the fast parallel run-length matrix computation algorithm described in section II-A can be employed. In the extreme case, only the first column of the matrix, the run-length-one vector, is used.

To further reduce the feature vector dimension and to decorrelate neighboring element values in the matrices, we use the principal component analysis method, also called Karhunen-Loeve Transform (KLT), and then use the Bhattacharyya distance measure to rank the eigenfeatures according to their discriminatory power.

Dominant principle component analysis:

To compute the Karhunen-Loeve Transform, let x_i be a feature vector sample. We form an n by m matrix

$$A = \begin{bmatrix} x_1(1) & x_2(1) & \ldots & x_m(1) \\ x_1(2) & x_2(2) & \ldots & x_m(2) \\ \ldots & \ldots & \ldots & \ldots \\ x_1(n) & x_2(n) & \ldots & x_m(n) \end{bmatrix} , \tag{22}$$

where n is the feature vector length and m is the number of training samples. The eigenvalues of the sample covariance matrix are computed in two ways, depending on the relative size of the feature vector and the

8

training sample number. If the feature vector length n is a small number, eigenvalues are computed by a standard procedure. The sample covariance matrix is estimated by

$$W = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)(x_i - \mu)^T = \frac{1}{m} A A^T, \tag{23}$$

where μ is the mean vector. The eigenvalues and eigenvectors are computed directly from W. However, for the feature vector formed by the four directional run-length matrices, n is a large number. For a neighborhood of 32x32 with 32 gray levels, n can reach a maximum of 4096. This means the covariance matrix is of size 4096x4096. Direct computation of the eigenvalues and eigenvectors becomes impractical. Fortunately, if the sample image number m is much smaller than n, the rank of W will only be m-1. A more efficient way to compute the eigenvectors is the dominant eigenvectors estimation method [11]. Consider the eigenvector e_i of $A^T A/m$, such that

$$\frac{1}{m} A^T A e_i = \lambda_i e_i. \tag{24}$$

By multiplying both sides by A, we have

$$\frac{1}{m} A A^T (A e_i) = \lambda_i (A e_i), \tag{25}$$

$$W(A e_i) = \lambda_i (A e_i). \tag{26}$$

This shows that $A e_i$ is the eigenvector of covariance matrix W. Therefore, we can compute the eigenvectors of a small m by m matrix $A^T A/m$, then calculate the first m eigenvectors of W as $A e_i$.

In the case where both m and n are large, we divide the training samples into $g = m/k$ groups of vectors,

$$A = \begin{bmatrix} \underbrace{\begin{matrix} x_1(1) \ \ldots \ x_k(1) \\ x_1(2) \ \ldots \ x_k(2) \\ \ldots \ \ldots \ \ldots \\ \ldots \ \ldots \ \ldots \\ x_1(n) \ \ldots \ x_k(n) \end{matrix}}_{A_1} & \underbrace{\begin{matrix} x_{k+1}(1) \ \ldots \ x_{2k}(1) \\ x_{k+1}(2) \ \ldots \ x_{2k}(2) \\ \ldots \ \ldots \ \ldots \\ \ldots \ \ldots \ \ldots \\ x_{k+1}(n) \ \ldots \ x_{2k}(n) \end{matrix}}_{A_2} \begin{matrix} \cdots \ \cdots \\ \cdots \ \cdots \\ \cdots \ \cdots \\ \cdots \ \cdots \end{matrix} & \underbrace{\begin{matrix} x_{(g-1)k+1}(1) \ \ldots \ x_m(1) \\ x_{(g-1)k+1}(2) \ \ldots \ x_m(2) \\ \ldots \ \ldots \ \ldots \\ \ldots \ \ldots \ \ldots \\ x_{(g-1)k+1}(n) \ \ldots \ x_m(n) \end{matrix}}_{A_g} \end{bmatrix}, \tag{27}$$

and apply the algorithm described above on each one of the g sample groups A_i. Then, the k dominant eigenvalues and eigenvectors are computed as the average of the computed g groups of eigenvalues and eigenvectors.

However, there are several implementation difficulties with this grouping approach. The number of samples in each group must be large enough and the samples must be uniformly selected from the whole data set to capture the dominant distribution directions of the original data set, so that the dominant eigenvec-

9

tors in each group approximate the dominant eigenvectors of the whole data set. Furthermore finding the corresponding eigenvectors among all groups is a nontrivial process.
Multi-level Dominant Eigenvector Estimation (MDEE) method.

To avoid these problems, we propose a new Multi-level Dominant Eigenvector Estimation method. Instead of grouping column vectors as in equation (27), we group the matrix in the row direction. By breaking the long feature vector into $g = n/k$ groups of small feature vectors of length k,

$$
A = \begin{bmatrix}
B_1 \left\{ \begin{bmatrix} x_1(1) \ x_2(1) \ \dots \ \dots \ x_m(1) \\ \dots \ \ \ \dots \ \ \ \dots \ \dots \ \ \dots \\ x_1(k) \ x_2(k) \ \dots \ \dots \ x_m(k) \end{bmatrix} \right\} \\
B_2 \left\{ \begin{bmatrix} x_1(k+1) \ x_2(k+1) \ \dots \ \dots \ x_m(k+1) \\ \dots \ \ \ \dots \ \ \ \dots \ \dots \ \ \dots \\ x_1(2k) \ \ \ x_2(2k) \ \ \dots \ \dots \ x_m(2k) \end{bmatrix} \right\} \\
\dots \ \ \dots \ \ \dots \ \dots \ \ \dots \\
\dots \ \ \dots \ \ \dots \ \dots \ \ \dots \\
B_g \left\{ \begin{bmatrix} x_1((g-1)k+1) \ x_2((g-1)k+1) \ \dots \ \dots \ x_m((g-1)k+1) \\ \dots \ \ \ \dots \ \ \ \dots \ \dots \ \ \dots \\ x_1(n) \ \ \ \ x_2(n) \ \ \ \dots \ \dots \ \ x_m(n) \end{bmatrix} \right\}
\end{bmatrix},
\tag{28}
$$

we can perform the KLT on each of the g group short feature vector set B_i. Then a new feature vector is formed by the first few selected dominant eigenfeatures of each group. The final eigenvectors are computed by applying the KLT to this new feature vector. To prove that the eigenvalues computed by MDEE are a close approximation of the standard KLT, we study the two-group case here. The feature vector matrix and its covariance matrix are

$$
A = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},
\tag{29}
$$

$$
W = AA^T = \begin{bmatrix} B_1B_1^T & B_1B_2^T \\ B_2B_1^T & B_2B_2^T \end{bmatrix} = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix}.
\tag{30}
$$

The averaging coefficients are omitted in the equations for simplicity. Let the eigenvector matrices of the covariance matrices W_1 and W_2 be T_1 and T_2 respectively, then

$$
T_1^T W_1 T_1 = \Lambda_1 ,
\tag{31}
$$

$$
T_2^T W_2 T_2 = \Lambda_2 .
\tag{32}
$$

where Λ_1 and Λ_2 are the diagonal eigenvalue matrices. The effective rotation matrix for the first-step group KLT is

$$T = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}. \tag{33}$$

T is also an orthogonal matrix, since

$$T^T T = \begin{bmatrix} T_1^T T_1 & 0 \\ 0 & T_2^T T_2 \end{bmatrix} = I. \tag{34}$$

So, after the first-step group KLT, the covariance matrix of the rotated feature vector,

$$W_r = T^T W T = \begin{bmatrix} \Lambda_1 & T_1^T W_{12} T_2 \\ T_2^T W_{21} T_1 & \Lambda_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \Lambda_{1b} & 0 \\ 0 & \Lambda_{1s} \end{bmatrix} & \begin{bmatrix} C_{bb} & C_{bs} \\ C_{sb} & C_{ss} \end{bmatrix}^T \\ \begin{bmatrix} C_{bb} & C_{bs} \\ C_{sb} & C_{ss} \end{bmatrix} & \begin{bmatrix} \Lambda_{2b} & 0 \\ 0 & \Lambda_{2s} \end{bmatrix} \end{bmatrix}, \tag{35}$$

is a similar matrix of the original feature vector covariance matrix W, because of the orthogonality of the rotation matrix T. Since similar matrices have the same eigenvalues, we can use the right most term of equation (35) to discuss the impact on W of keeping only the first few dominant eigenvalues in each group. In equation (35), Λ_{nb} and Λ_{ns} represent the larger dominant eigenvalue section and the smaller negligible eigenvalue section of the eigenvalue matrix Λ_n respectively, for $n = 1$ or 2. C_{xx}, where $x = b$ or s, represents the cross-covariance matrix of the two groups of rotated features. By keeping only the dominant eigenvalues, the new feature vector covariance matrix becomes

$$W_d = \begin{bmatrix} \Lambda_{1b} & C_{bb}^T \\ C_{bb} & \Lambda_{2b} \end{bmatrix}. \tag{36}$$

The terms removed from W_r are Λ_{1s}, Λ_{2s}, C_{ss}, C_{bs} and C_{sb}. Since most energy is contained in the dominant eigenvalues, the loss of information due to Λ_{1s} and Λ_{2s} should be very small. The energy contained in the cross-covariance matrix of the two small energy feature vectors, C_{ss}, should therefore be even smaller.

We can also show that C_{bs} and C_{sb} cannot be large either. If the two group features B_1 and B_2 are fairly uncorrelated with each other, then all the cross-covariance C_{xx} matrices in (35) will be very small. On the other hand, if the two group features are strongly correlated with each other, the dominant eigenfeatures of the two group will be very similar. Therefore the cross-covariance matrix C_{bs} of group-two large features with group-one small features will be similar to the cross-covariance matrix of the group-one large features with group-one small features, which is zero due to the decorrelation property of the KLT transform.

When the two group features B_1 and B_2 are partially correlated, the correlated part should be mostly signal, since noise parts of the variable B_1 and B_2 rarely correlate with each other. The basic property of the KLT is to preserve all signal energy in the first few large eigenvalues. Therefore, most signal energy in B_2, and especially most of the B_2 signal energy that is correlated with B_1, will be preserved in the large eigenvalue section of B_2 covariance matrix. The energy that is discarded in the small eigenvalue section of B_2 will contain little if any energy that is correlated with B_1. Therefore, C_{bs} and C_{sb} should be very small, and we will not lose much information by removing them from the covariance matrix W_r.

Now that we have shown that the covariance matrix W_d is a close approximation of W_r, and W_r is a similar matrix of W, we can say that the eigenvalues from W_d, i.e., by the MDEE method, are indeed a close approximation of the eigenvalues computed from W, i.e., by the standard KLT method.

Significant reduction of computational time can be achieved by the MDEE over the standard KLT. For example, if a feature vector of length $n = 1000$ is broken into 10 vector groups of length 100, and 10% of the eigenfeatures in each group are saved for the second-level eigenvalue computation, the computational complexity for the MDEE is $11(n/10)^3$, which is nearly two orders of magnitude faster than the KLT's 1000^3. Furthermore, the algorithm offers an excellent opportunity for parallel computation. If all individual group KLTs are computed in parallel, a near three-order-of-magnitude speed increase can be achieved for this example.

However, it is well known that the KLT features are optimal for data representation but not necessarily the best for discrimination. To measure the class separability of each feature, some other criterion must be employed. We choose the Bhattacharyya distance measure.

Bhattacharyya Distance Measure:

We select the Bhattacharyya distance in this work because it has a direct relation with the error bound of the Gaussian classifier and has a simple form for features with normal distributions. As indicated by Fukunaga [11], for a two-classes problem

$$\varepsilon_{(c_1, c_2)} \leq [P(c_1)P(c_2)]^{\frac{1}{2}} \exp[-\beta_{d(c_1, c_2)}], \tag{37}$$

where $P(c_i)$ is the prior probability of class c_i, ε is the probability of error for a Gaussian classifier and β_d is the Bhattacharyya distance. Because its inverse gives the upper bound on the probability of error, β_d can be an effective measure of class separability. For a normal distribution, β_d has the analytical form

$$\beta_{d(c_1, c_2)} = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{W_1 + W_2}{2}\right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{\left|\frac{1}{2}(W_1 + W_2)\right|}{|W_1|^{1/2}|W_2|^{1/2}}. \tag{38}$$

where μ_1, μ_2 and W_1, W_2 are the mean vectors and covariance matrices of the two class distributions. The many possible combinations of several features and the possibility of covariance matrix singularity make it impractical to compute the Bhattacharyya distance for several features at once. The one-at-a-time method

is adopted instead. The formula is the same as equation (38), only with the covariance matrix W replaced by the variance and the mean vector μ replaced by the class mean. As for multi-class problems, the overall probability of error can be bounded by [19]

$$\varepsilon \le \sum_{i>j}^{M} \sum_{j=1}^{M} \varepsilon_{(c_i, c_j)}, \tag{39}$$

where ε and $\varepsilon_{(ci,cj)}$ ($i, j = 1,2,..., M$) are the probabilities of overall error and the pair-wise error between class i and j respectively. From Equations (37) and (39) we select the features according to the minimum total upper error bound. Because the test data size is the same for all classes in our experiment, the prior probabilities $P(c_i)$ are equal for all classes. Thus, we select features with small values of

$$S_b = \sum_{i>j}^{M} \sum_{j=1}^{M} \exp[-\beta_{d(c_i, c_j)}]. \tag{40}$$

Throughout the experiments in section III, we select the first 30 features with largest eigenvalues, rank these KLT-decorrelated features by their S_b values, and use the first n features with the smallest S_b for classification. We run the feature length n from 1 to 30 to select the one that gives the best performance as the final feature vector length. This is apparently not an optimal searching approach, since a combination of the first n best individual features may not be the best length n feature vector. However, the experimental results suggest it to be a close approximation. Since all features are first decorrelated by the KLT transform, as we increase the feature length each additional feature brings in new uncorrelated information and noise. When their S_b values increase to a certain point, the new features start to bring in more noise than information, suggesting that a suboptimal feature length is reached. The experiments show that most best feature lengths are between 10 and 20.

D. Classification algorithm

Since the main focus of this work is the feature extraction algorithm, we use a simple Gaussian classifier for the experiments. Let the class mean and covariance matrix of the feature vector x be m_i and W_i respectively, a distance measure is defined as [26]

$$D_i = (x - \mu_i)^T W_i^{-1} (x - \mu_i) + \ln|W_i|, \tag{41}$$

where the first term on the right of the equation is actually the Mahalanobis distance. The decision rule is

$$x \in C_L \quad \text{when } D_L = min\{D_i\}. \tag{42}$$

III. EXPERIMENTS AND DISCUSSION

In this section, two separate data sets are used for the texture classification experiment. We first make detail comparison between various DRM features and the traditional run-length features on the classification of eight Brodatz images. We then compare the best DRM features with the co-occurrence

features and the wavelet features on the classification of a larger data set -- sixteen Vistex images.

A. Data description

The first data set comprises the eight Brodatz images [2], which are shown in Fig. 4. Each image is of size 256x256 with 256 gray levels. The images are first quantized into 32 gray levels using equal-probability quantization. Each class is divided into 225 sample images of dimension 32x32 with fifty percent overlapping. Sixty samples of each class are used as training data, so the training data size is 480 samples and the testing data size is 1320.

As we will see from the result on the above data set, most of our new algorithms give perfect classification. To further compare the performance of these new algorithms and their consistency when applied to a larger natural image set, we conducted a second experiment on a set of sixteen images from Vistex texture image database established by the MIT Media Lab. Unlike Brodatz images which are mostly obtained in well controlled studio conditions, the Vistex images were taken under natural lighting conditions. They pose a more realistic challenge for texture classification algorithms. Table 1 is the description of the sixteen Vistex images shown in Fig.5. The same 32 gray level quantization is applied to each image. This quantization will make all the image classes have the same flat histogram, indistinguishable by mean and variance features. However, unlike most texture classification experiments, no adaptive histogram equalization is applied to the images to compensate the nonuniform lighting. This makes the classification more difficult and the classification result a closer reflection of real-world applications. Each class is again divided into 225 samples of dimension 32x32 with fifty percent overlapping. Sixty samples of each class are used as training data. So the training data has 1920 samples and the testing data has 2640 samples.

B. Classification using the traditional run-length features

Table 2 shows the classification results using the traditional run-length features directly on the Brodatz images. Similar to [7], the feature groups tested are the original five features of Galloway [12], the two features of Chu et al. [4], and the four new features of Dasarathy and Holder [7]. All four-direction features are used. Contrary to the good classification results on only four classes of 80 samples in [7], all groups of features perform poorly here. With only 35% classification accuracy, the result of using all three group features together is much worse than any single group features. However, by applying the feature selection algorithms, i.e., KLT plus Bhattacharyya distance measure, to the feature vector before classification, improved results are shown in Table 3. In this case, the feature vector containing all three group features achieves 88% accuracy, far better than any single group features. This is mainly because of the close correlation of the three groups of features.

To see the degree of correlation, we compute the auto-correlation coefficient matrix of the complete run-length feature vector shown in Fig. 6. Many coefficient values in the matrix are close to one and the high correlations can also be seen in the scatter plots of several strongly correlated features, as illustrated in Fig. 7. The poor classification performance of correlated features indicates that additional features bring in a great deal of noise, which overwhelms any marginal benefit of mostly redundant information contained in the added features. This shows the importance of using the KLT transform to extract

decorrelated information.

C. Classification using the new DRM features

Figures 8 and 9 show the scatter plots of the top eight features obtained by applying the MDEE transform on the original run-length matrix and on the run-length-one vector, respectively. Almost perfectly separable clustering can be seen for most of the eight image classes in both cases, in sharp contrast to the overlapping clusters in Fig. 7 using the traditional feature vector.

The classification results using the DRM features are summarized in Table 4. Notice the dramatic reduction of feature length from several hundreds to around ten, comparable with the traditional feature vector length. The results indicate that a compact, optimal run-length feature vector can be extracted by the MDEE method, without resort to *ad hoc* functions.

With only such a small number of features, perfect classification is achieved with the original matrix and with most of the new matrices and vectors. The only exceptions in Table 4 are the RLRN vector and the long-run region of the run-length matrix. The poor performance of the long-run region matrix and the good performance of the short-run region matrix indicate that most texture information is indeed concentrated in the short-run region. This also helps to explain the poor performance of the RLRN vector. Since most information is stored in the first few columns of the run-length matrix, the only important features in RLRN are the first few features, which are the summation of the first few columns. The gray level information is totally lost.

D. Comparison with other methods

We now compare the new run-length method with the widely used co-occurrence method and the recently proposed wavelet method on a larger and more difficult Vistex data set. For the co-occurrence method, thirteen co-occurrence features--Contrast, Correlation, Entropy, and Variance, etc.--are computed for each of the four directions as described in [14]; for the wavelet method, the texture feature used for each wavelet decomposition channel is the energy feature:

$$ENG = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x(i,j) - \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}x(i,j)\right)^2, \tag{43}$$

where $x(i, j)$ denotes an element of the wavelet packet coefficient in each frequency channel and M and N are the size of the channel. The same feature selection method in section II-C is applied to the co-occurrence and wavelet feature vectors.

The classification results on the sixteen Vistex images using various DRM features are first shown in Table 5. About 97% classification accuracy is achieved by most feature vectors. An especially interesting result is that the run-length-one vector gives excellent performance, similar to that of the original full matrix. This confirms that the fast, parallel processing algorithm can be used to extract useful run-length texture features.

Classification results using co-occurrence and wavelet features on the sixteen Vistex images are shown

15

in Table 6. From the results, we can see that the run-length features are no longer the least effective features. In fact, the run-length features perform comparably with the co-occurrence features and better than the wavelet features. This demonstrates that there is rich texture information contained in the run-length matrices and that a method of extracting such information is of paramount importance to successful classification.

The poor results of the wavelet features are inconsistent with several previous studies [3] [18], where wavelet features generate near perfect classifications. This is mainly because that we use a much smaller texture sample size, 32x32, than the ones used in most previous studies, 64x64 or 128x128 [3] [18]. Such a small image size may not be enough to estimate a stable frequency energy distribution. However, it is important for any texture classification algorithm to give good performance on small size images, so that they can be useful for more difficult image segmentation applications.

To confirm this sample size effect, we divide each Vistex image class into 169 sample images of dimension 64x64 with 75% overlapping between neighborhood samples. Only 39 samples in each class are used as training data, so the training data size is 624 samples and the testing data size is 2080 samples. Table 7 shows the classification results. Near perfect classifications are achieved by all three methods, similar to results in [3] [18]. As we increase the training data size to 1456 samples and decrease testing data size to 1248, all three feature vectors produce perfect classifications, as shown in Table 8.

IV. CONCLUSION

In this paper, we extract a new set of run-length texture features that significantly improve image classification accuracy over traditional run-length features. By directly using part or all of the run-length matrix as a feature vector, much of the texture information is preserved. This approach is made possible by the introduction of a new multi-level dominant eigenvector estimation method. The MDEE reduces the computation complexity of KLT by several orders of magnitude. Combined with the Bhattacharyya distance measure, they form an efficient feature selection algorithm.

The advantage of this approach is demonstrated experimentally by the classification of two independent texture data sets. Perfect classification is achieved on the eight Brodatz images. The 97% classification accuracy on the sixteen Vistex images further confirms the effectiveness of the algorithm. Experimentally, we observe that most texture information is stored in the first few columns of the run-length matrix, especially in the first column. This observation justifies development of a new, fast, parallel run-length matrix computation scheme.

Comparisons of this new approach with the co-occurrence and wavelet methods demonstrate that the run-length matrices possesses as much discriminatory information as these successful conventional texture features and that a good method of extracting such information is key to the success of the classification. We are currently investigating the application of the new feature extraction approach on other texture matrices. We hope our work here will also renew the interest in run-length texture features, and will promote more future applications.

16

REFERENCES

[1]   F. Ade, "Characterization of textures by `eigenfilters'," *Signal Processing*, vol. 5, pp. 451-457, 1983.

[2]   P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York, 1966.

[3]   T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Processing*, vol. 2, pp 429-441, Oct. 1993.

[4]   A. Chu, C. M. Sehgal and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognition Letters*, vol. 11, pp 415-420. June 1990.

[5]   R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, pp. 713-719, Mar. 1992.

[6]   R. W. Conners and C. A. Harlow, "A Theoretical Comparison of Texture Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, No. 3, pp. 204-222, May 1980.

[7]   B. R. Dasarathy and E. B. Holder, "Image characterizations based on joint gray-level run-length distributions," *Pattern Recognition Letters*, vol. 12, pp. 497-502, 1991.

[8]   I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol 41, pp. 909-996, 1988.

[9]   D. Dunn, W. E. Higgins and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 16, pp. 130-149, Feb. 1994.

[10] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biol. Cybern.*, vol. 61, pp. 103-113, 1989.

[11] K. Fukunaga, *Introduction to statistical pattern recognition.* New York: Academic Press, 1972.

[12] M. M. Galloway, "Texture analysis using gray level run lengths," *Comput. Graphics Image Processing*, vol. 4, pp. 172-179, June 1975.

[13] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp 786-804, May 1979.

[14] R. M. Haralick, K. S. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610-621, 1973.

[15] M. B. Henke-Reed and S. N. C. Cheng, "Cloth texture classification using the wavelet transform," *Journal of Imaging Science and Technology*, vol. 37, pp. 610, Dec. 1993.

17

[16] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," in *Proc. of the IEEE Inter. Conf. on Sys. Man and Cyber.*, pp. 14-19, Los Angeles, CA, 1990.

[17] M. E. Jernigan and F. D'Astous, "Entropy-based texture analysis in the spatial frequency domain," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 6 pp. 237-243, March 1984.

[18] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 15 pp. 1186-1191, Nov. 1993.

[19] D. G. Lainiotis, "A class of upper bounds on probability of error for multihypotheses pattern recognition," *IEEE Trans. Information Theory* IT-15, pp. 730-731, 1969.

[20] K. I. Laws, "Textured Image Segmentation," Ph.D. dissertation, Image Processing Inst., Univ. of Southern California, 1980.

[21] S. Liu and M. E. Jernigan, "Texture analysis and discrimination in additive noise," *Computer Vision, Graphics, and Image Processing*, vol. 49, pp. 52-67, 1990.

[22] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Opt. Soc. Am.* A, Vol. 7, No. 5, pp 923-932, 1990.

[23] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 11 pp. 674-693, July 1989.

[24] S.G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 2091-2110, Dec. 1989.

[25] Y. Meyer, *Wavelets: Algorithms and Applications*, SIAM, Philadelphia, PA, 1993, Translated and revised by R. D. Ryan.

[26] C. W. Therrien, *Decision Estimation and Classification, An Introduction to Pattern Recognition and Related Topics*, Wiley, New York, 1989.

[27] M. Unser, "Local linear transforms for texture measurements," *Signal Processing*, Vol. 11, No. 1, July 1986, pp. 61-79.

[28] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 11, no. 7, pp. 717-728, July 1989.

[29] H. Wechsler, "Texture analysis - a survey," *Signal Processing*, vol. 2, pp. 271-282, 1980.

[30] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, No. 4, pp. 269-285, 1976.
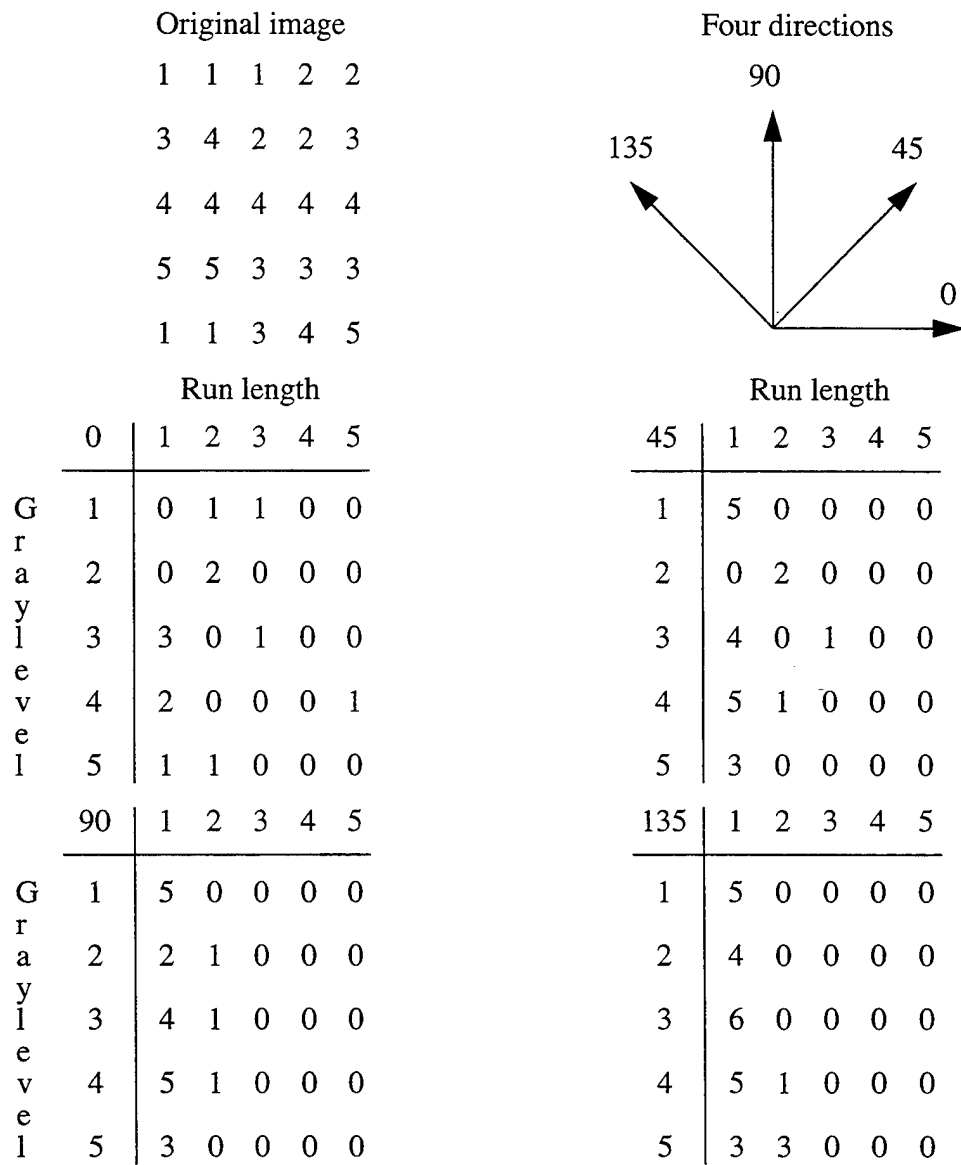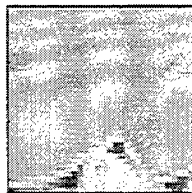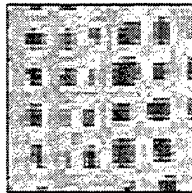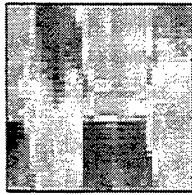
Original image

1	1	1	2	2
3	4	2	2	3
4	4	4	4	4
5	5	3	3	3
1	1	3	4	5

Run length

0	1	2	3	4	5
G 1	0	1	1	0	0
r					
a 2	0	2	0	0	0
y					
l 3	3	0	1	0	0
e					
v 4	2	0	0	0	1
e					
l 5	1	1	0	0	0

90	1	2	3	4	5
G 1	5	0	0	0	0
r					
a 2	2	1	0	0	0
y					
l 3	4	1	0	0	0
e					
v 4	5	1	0	0	0
e					
l 5	3	0	0	0	0

Four directions

90

135   45

0

Run length

45	1	2	3	4	5
1	5	0	0	0	0
2	0	2	0	0	0
3	4	0	1	0	0
4	5	1	0	0	0
5	3	0	0	0	0

135	1	2	3	4	5
1	5	0	0	0	0
2	4	0	0	0	0
3	6	0	0	0	0
4	5	1	0	0	0
5	3	3	0	0	0

Fig. 1. Four directional gray-level run-length matrices.

Texture image samples   Runlength matrices in four directions
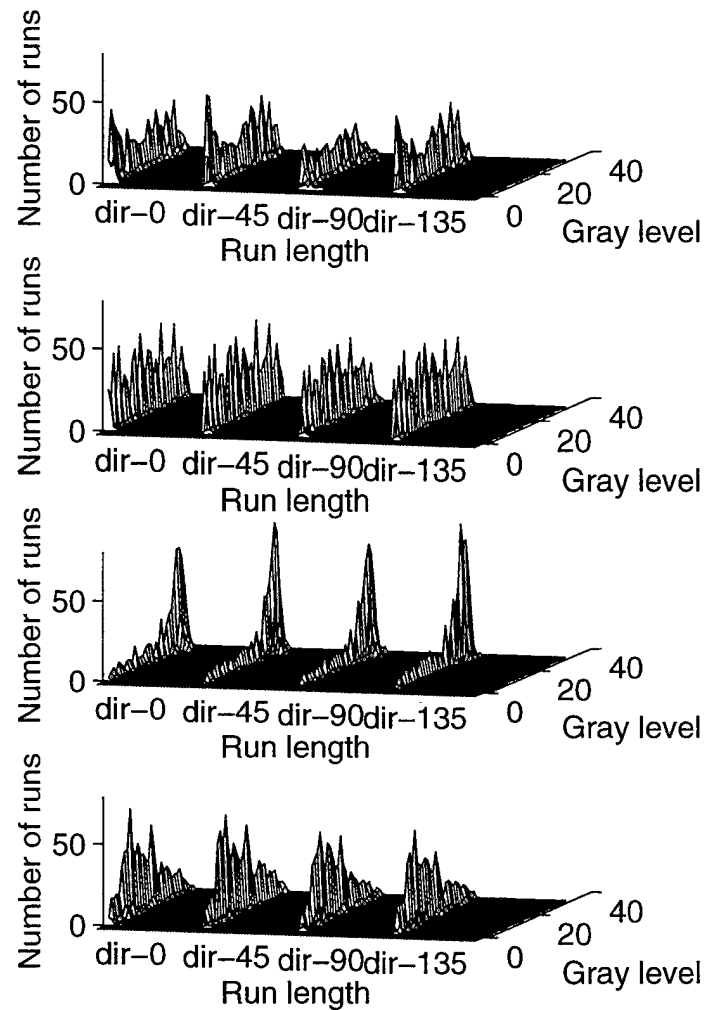
Fig. 2. The four directional run-length matrices of several Brodatz texture samples. Each
image sample is of size 32x32 with 32 gray levels. The four directional (0, 45, 90 and
135 degree directions) run-length matrices are combined into a single matrix. The left-
most column of each directional matrix is the run-length-one vector, which has much
larger values than the other columns.

Fig. 3. Run-emphasis regions of several traditional run-length texture features.
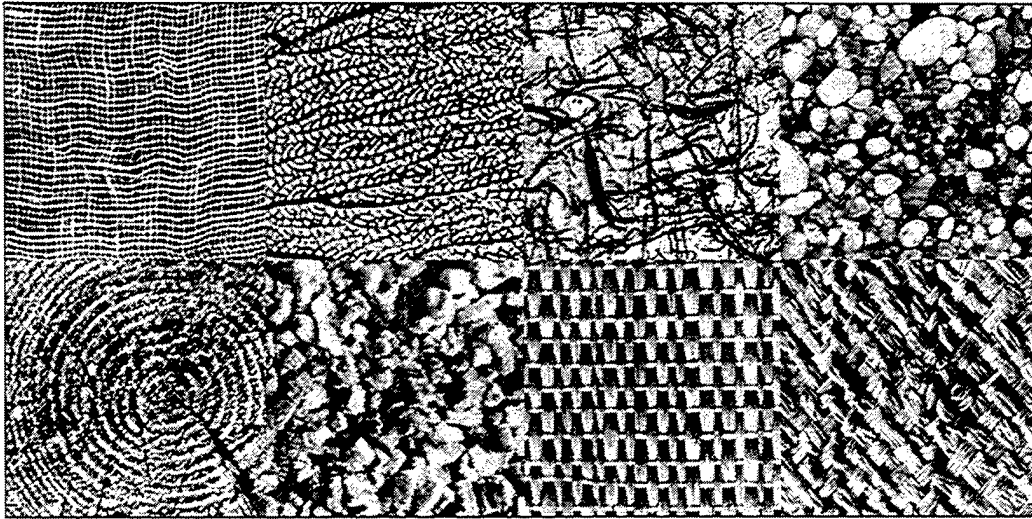
Fig. 4. Eight Brodatz textures. Row 1: burlap, seafan, ricepaper, pebbles23; Row 2: tree, mica, straw, raffia.
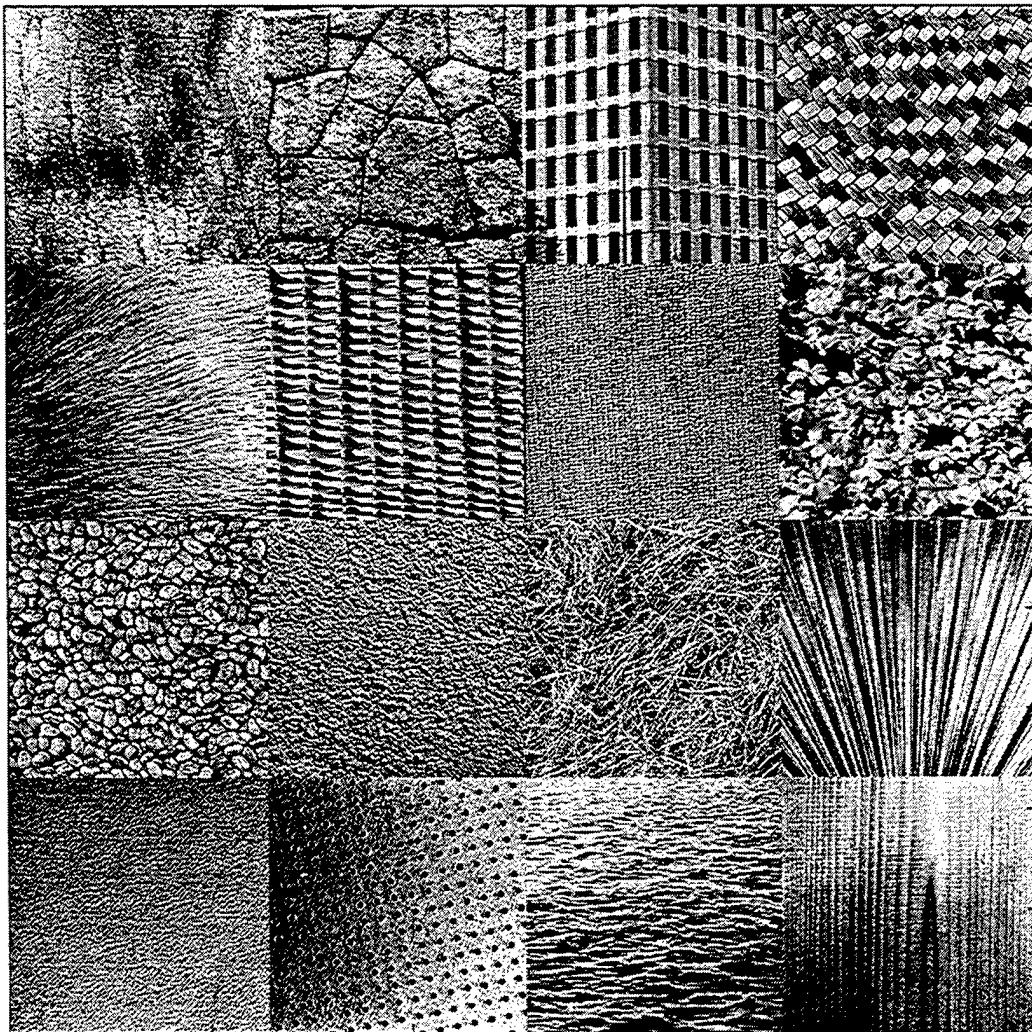
Fig. 5. Sixteen Vistex textures. Descriptions are in Table 1.
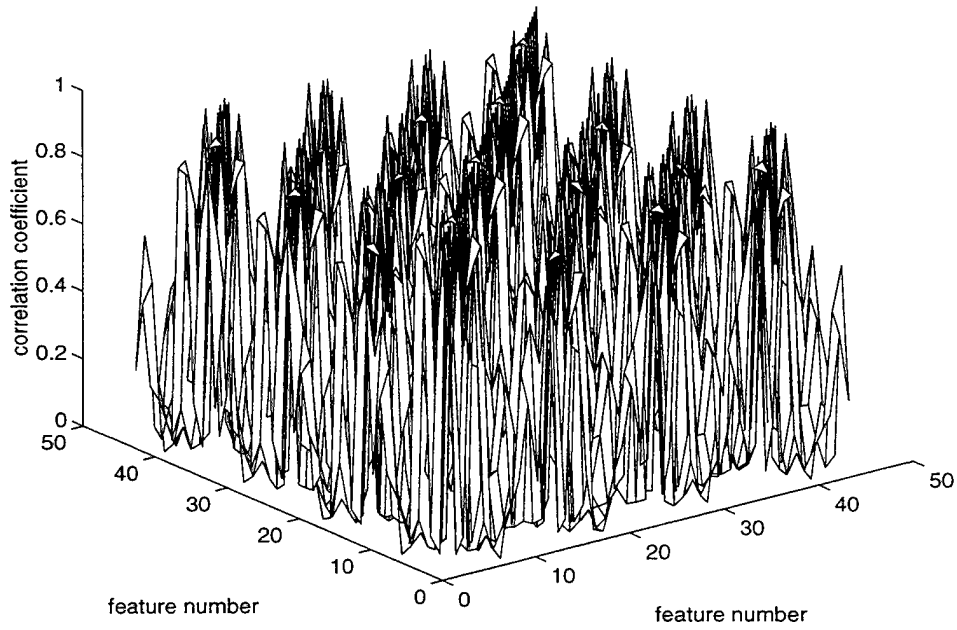
Fig. 6. Auto-correlation coefficient matrix of the traditional run-length features.
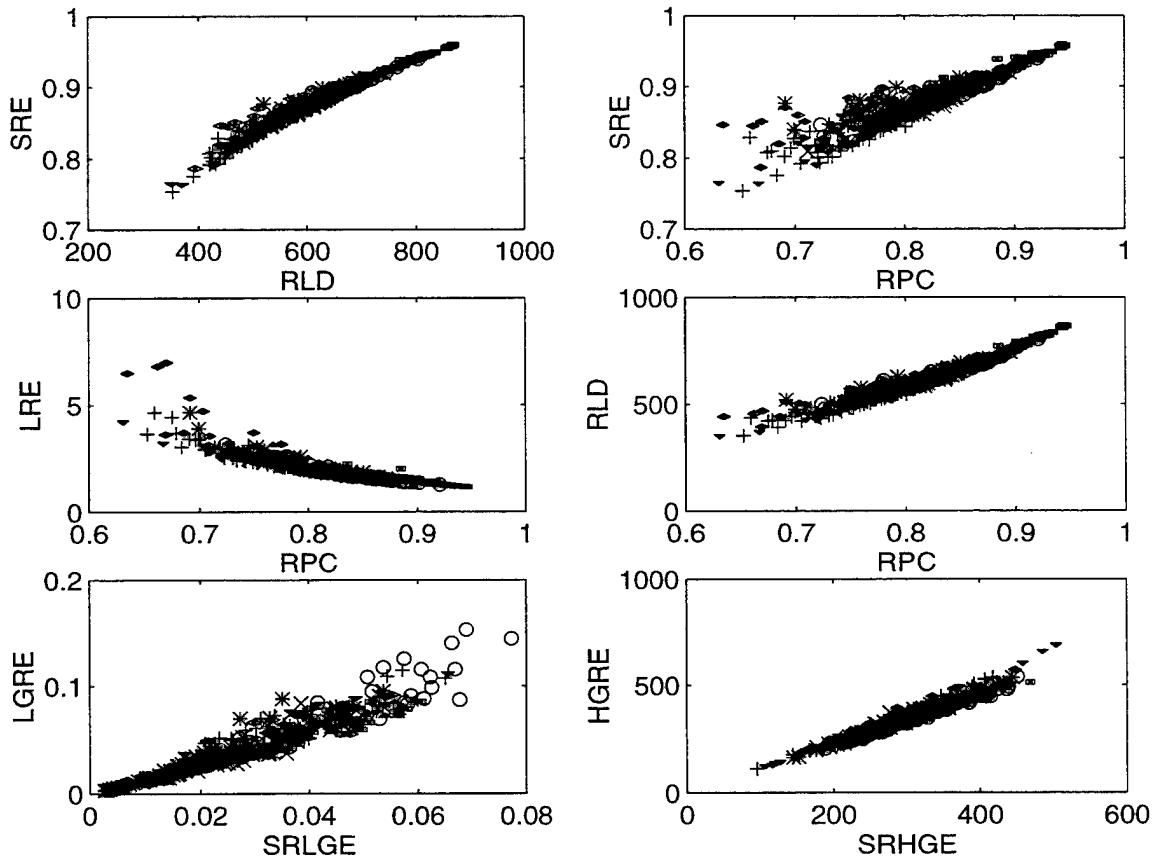
Fig. 7. Scatter plots of several highly correlated traditional run-length texture features of the eight Brodatz textures. Due to overlap, not all eight class symbols can be discerned.
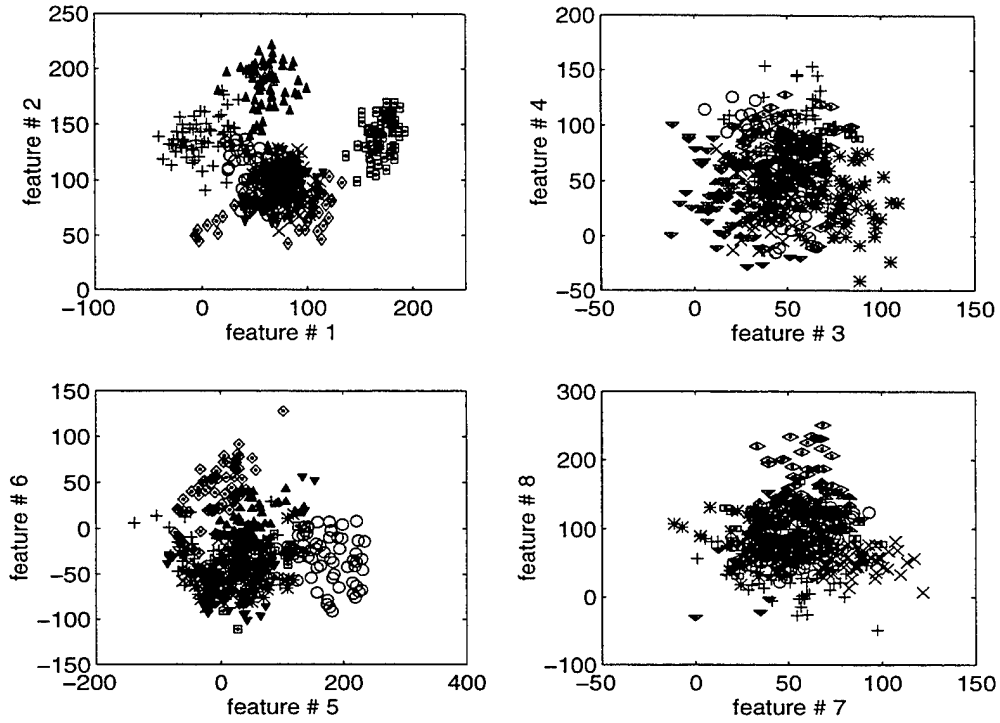
22

Fig. 8. Scatter plots of the top eight features extracted by applying a MDEE transform on the original run-length matrices of the Brodatz textures. Linearly separable clustering is observed for most of the eight texture classes.
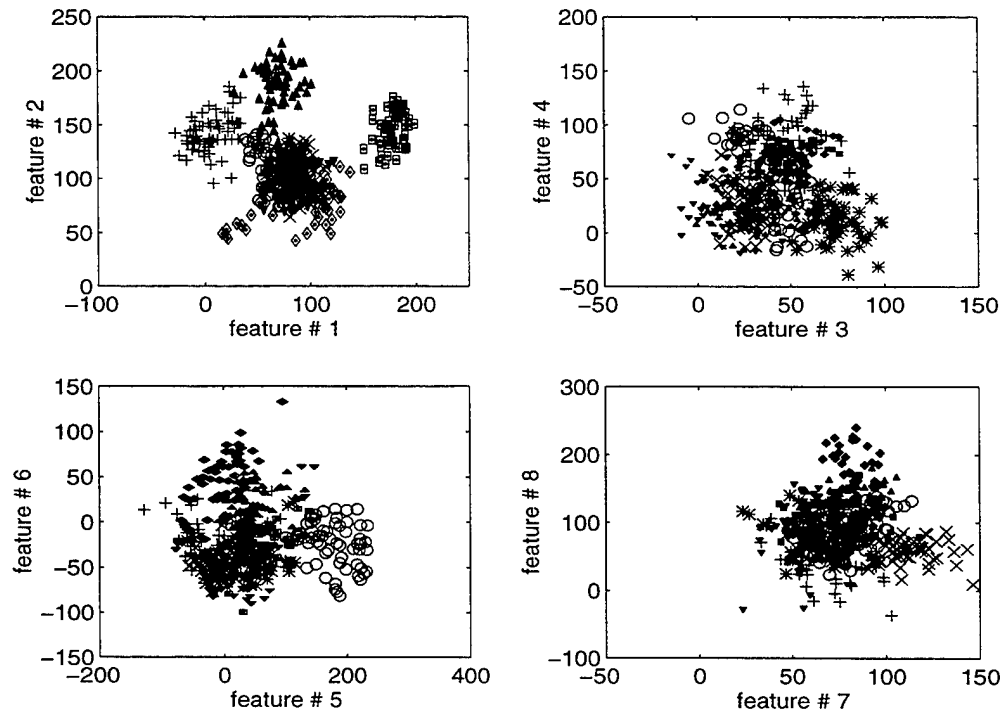
Fig. 9. Scatter plots of the top eight features extracted by applying a MDEE transform on the run-length-one vector of the Brodatz textures. Linearly separable clustering is observed for most of the eight texture classes.

Table 1: Vistex texture images description.

Image name	Contents	Lighting	Perspective
Bark.0008	tree bark	daylight direct right	frontal plane
Brick.0004	brick	daylight indirect right	frontal plane
Buildings.0009	building	daylight indirect	oblique
Fabric.0001	straw rattan	artificial incandescent	frontal plane
Fabric.0005	fur	artificial incandescent	frontal plane
Fabric.0013	wicker	daylight indirect	frontal plane
Fabric.0017.	carpet backing	daylight indirect	frontal plane
Flowers.0007	flowers	daylight direct	frontal plane
Food.0000	lima beans	artificial incandescent	frontal plane
Food.0005	coffee grounds	artificial strobe	frontal plane
Grass.0002	grass straw	daylight direct	frontal plane
Leaves.0002	plant leaf	daylight direct	frontal plane
Metal.0001	metal reflector sheet	artificial strobe	frontal plane
Tile.0007	ceiling tile	artificial strobe	frontal plane
Water.0006	water	daylight direct	oblique
Wood.0002	wood	daylight indirect	frontal plane

Table 2: Brodatz texture classification results using the traditional run-length features.

Feature name	Original feature length	Number of selected features	Correct classification rate		
			Training data	Testing data	All data
G5	20	20	64.6	60.7	61.7
C2	8	8	61.2	41.8	47.0
D4	16	16	84.4	59.1	65.8
ALL	44	44	35.6	35.4	35.4

Table 3: Brodatz texture classification results using the new feature selection method on the traditional run-length features.

Feature name	Original feature length	Number of selected features	Correct classification rate		
			Training data	Testing data	All data
G5	20	12	88.5	74.9	78.6
C2	8	8	61.2	41.8	47.0
D4	16	16	84.4	59.1	65.8
ALL	44	24	99.4	83.7	87.9

Table 4: Brodatz texture classification results using the new dominant run-length matrix features.

Feature name	Original feature length	Number of selected features	Correct classification rate		
			Training data	Testing data	All data
p: columns 1:4	512	11	100.0	100.0	100.0
p: columns 5:32	3584	8	53.3	41.3	44.5
p: whole matrix	4096	11	100.0	100.0	100.0
p_p: columns 1:4	512	7	100.0	100.0	100.0
p_p: columns 5:32	3584	17	69.6	41.4	48.9
p_p: whole matrix	4096	10	100.0	100.0	100.0
p_g: GLRNV	128	8	100.0	100.0	100.0
p_r: RLRNV	128	20	95.2	63.9	72.3
p_o: GLRLOV	128	11	100.0	100.0	100.0

Table 5: Vistex texture classification results using the new dominant run-length matrix features.

Feature name	Original feature length	Number of selected features	Correct classification rate		
			Training data	Testing data	All data
p: columns 1:4	512	17	99.9	96.8	97.6
p: whole matrix	4096	18	99.9	98.0	98.5
p_p: columns 1:4	512	19	100.0	96.8	97.6
p_p: whole matrix	4096	24	100.0	97.5	98.1
p_g: GLRNV	128	23	100.0	93.9	95.6
p_o: GLRLOV	128	18	99.8	97.0	97.8

Table 6: Vistex Texture classification results using the co-occurrence, the wavelet, and the new run-length features.

Feature name		Original feature length	Number of selected features	Correct classification rate		
				Training data	Testing data	All data
Co-occurrence		52	29	100.0	97.4	98.1
Wavelet	Level 2	16	13	98.2	90.6	92.7
	Level 3	64	20	98.6	90.1	92.4
	All Levels	84	15	97.9	90.6	92.5
Run-length		4096	18	99.9	98.0	98.5

26

Table 7: Vistex Texture classification results using the co-occurrence, the wavelet, and the run-length features, with image sample size 64x64, and training image # : testing image # = 624:2080.

Feature name		Original feature length	Number of selected features	Correct classification rate		
				Training data	Testing data	All data
Co-occurrence		52	24	100.0	99.8	99.8
Wavelet	Level 2	16	10	100.0	99.5	99.6
	Level 3	64	12	100.0	99.3	99.5
	Level 4	256	22	100.0	98.2	98.6
	All Levels	340	24	100.0	98.1	98.5
Run-length		8192	13	100.0	100.0	100.0

Table 8: Vistex Texture classification results using the co-occurrence, the wavelet, and the run-length features, with image sample size 64x64, and training image # : testing image # = 1456:1248.

Feature name		Original feature length	Number of selected features	Correct classification rate		
				Training data	Testing data	All data
Co-occurrence		52	13	100.0	100.0	100.0
Wavelet	Level 2	16	10	99.9	100.0	100.0
	Level 3	64	15	100.0	99.9	100.0
	Level 4	256	24	100.0	99.9	100.0
	All Levels	340	27	100.0	100.0	100.0
Run-length		8192	11	100.0	100.0	100.0

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-97-07	2.	3. Recipient's Accession No.

4. Title and Subtitle
Dominant Run-Length Method for Image Classification

5. Report Date
June 1997

6.

7. Author(s) Xiaoou Tang

8. Performing Organization Rept. No.
WHOI-97-07

9. Performing Organization Name and Address

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.

(C)   N00014-93-1-0602

(G)

12. Sponsoring Organization Name and Address
Office of Naval Research

13. Type of Report & Period Covered
Technical Report

14.

15. Supplementary Notes
This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-97-07.

16. Abstract (Limit: 200 words)

In this paper, we develop a new run-length texture feature extraction algorithm that significantly improves image classification accuracy over traditional techniques. By directly using part or all of the run-length matrix as a feature vector, much of the texture information is preserved. This approach is made possible by the introduction of a new multi-level dominant eigenvector estimation algorithm. It reduces the computational complexity of the Karhunen-Loeve Transform by several orders of magnitude. Combined with the Bhattacharyya distance measure, they form an efficient feature selection algorithm. The advantage of this approach is demonstrated experimentally by the classification of two independent texture data sets. Perfect classification is achieved on the first data set of eight Brodatz textures. The 97% classification accuracy on the second data set of sixteen Vistex images further confirms the effectiveness of the algorithm. Based on the observation that most texture information is contained in the first few columns of the run-length matrix, especially in the first column, we develop a new fast, parallel run-length matrix computation scheme. Comparisons with the co-occurrence and wavelet methods demonstrate that the run-length matrices contain great discriminatory information and that a method of extracting such information is of paramount importance to successful classification.

17. Document Analysis   a. Descriptors
Texture Image Classification
Run Length
Karhunen Loeve Transform

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 27
Approved for public release; distribution unlimited.	20. Security Class (This Page)	22. Price

(See ANSI-Z39.18)   *See Instructions on Reverse*

OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce